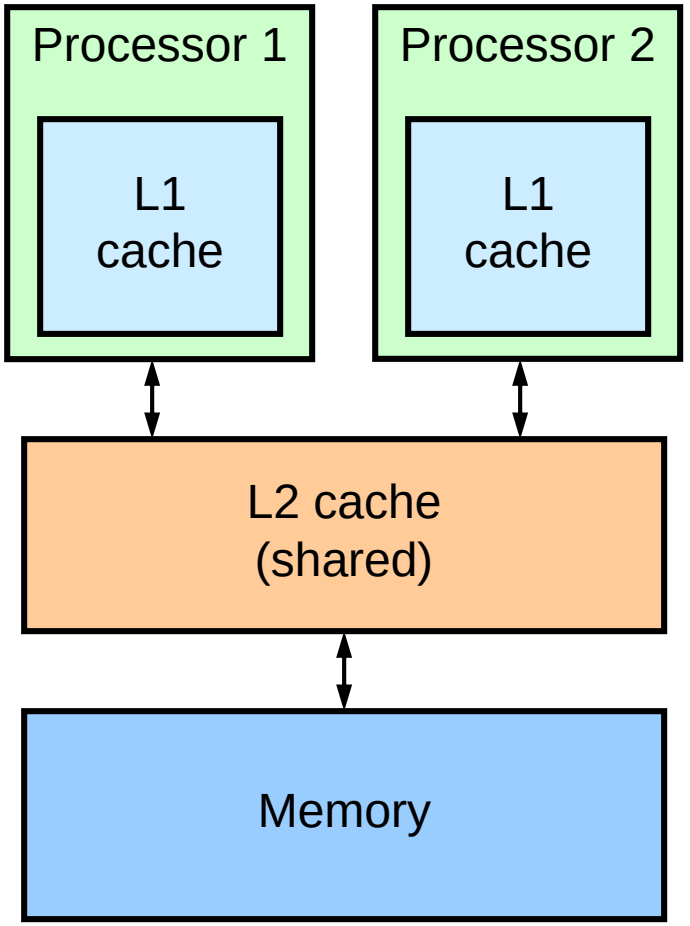
# MESI Protocol

# Multi-processor System

- **A memory system is *coherent* if**
  1. If P1 writes to address X,
     and later on P2 reads X,
     and there are no other writes to X in between
     ⇒ P2's read returns the value written by P1's write
  2. Writes to the same location are serialized:
     two writes to location X are seen in the same order by all processors

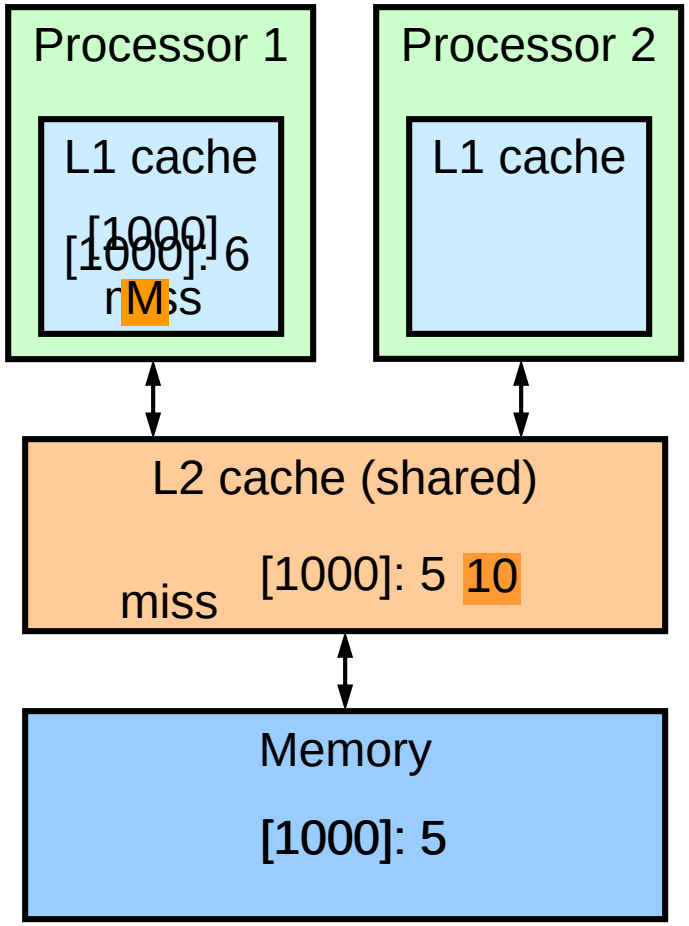


# MESI Protocol

- **Each cache line can be in one of 4 states**
  - Invalid – Line's data is not valid
  - Shared – Line is valid and not dirty,
    copies may exist in other processors
  - Exclusive – Line is valid and not dirty,
    other processors do not have the line in their local caches
  - Modified – Line is valid and dirty,
    other processors do not have the line in their local caches

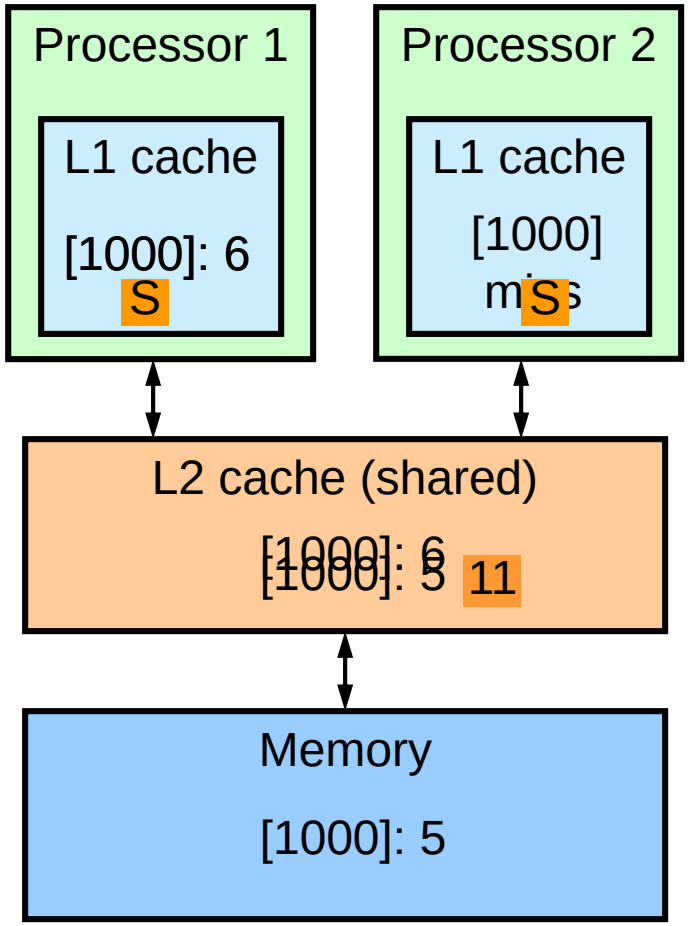
# Multi-processor System: Example

- **P1 reads 1000**
- **P1 writes 1000**



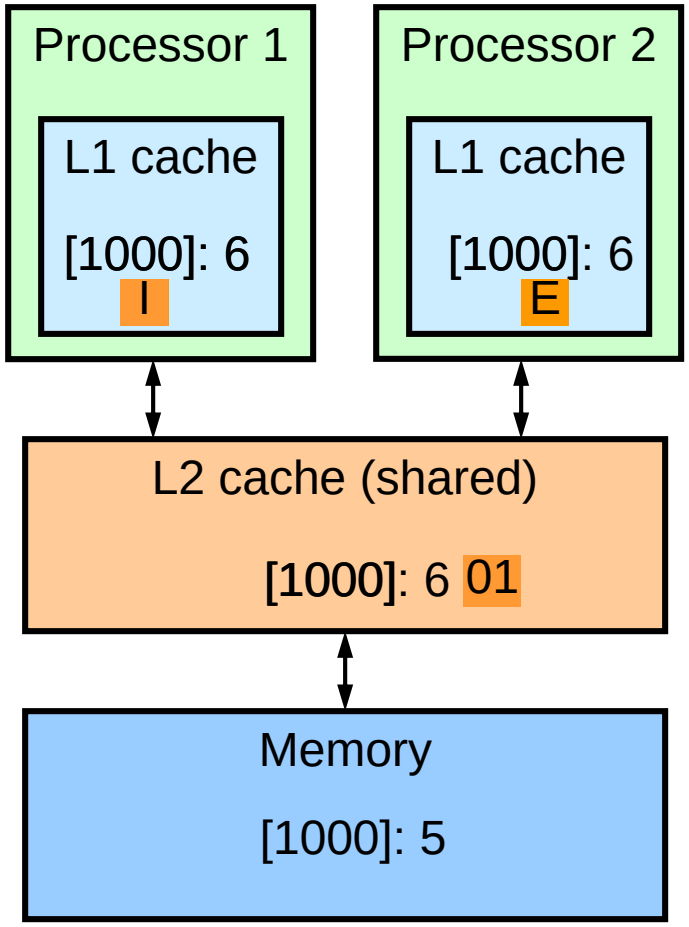
# Multi-processor System: Example

- P1 reads 1000
- P1 writes 1000
- P2 reads 1000
- L2 snoops 1000
- P1 writes back 1000
- P2 gets 1000



# Multi-processor System: Example

- P1 reads 1000
- P1 writes 1000
- P2 reads 1000
- L2 snoops 1000
- P1 writes back 1000
- P2 gets 1000
- P2 requests for ownership with write intent



# Core Valid Bits and Inclusion

- **L2 keeps track of the presence of each line in each of the Core's L1 caches**
  - Determine if it needs to send a snoop to a processor
  - Determine in what state to provide a requested line (S,E)
  - Maintain Core Valid Bits (CVB) per cache line
  - ⇒ Need to guarantee that the L1 caches in each Core are inclusive of the L2 cache
- **When L2 evicts a line**
  - L2 sends a snoop invalidate to all processors that have it
  - If the line is modified in the L1 cache of one of the processors (in which case it exist only in that processor)
    - The processor responds by sending the updated value to L2
    - When the line is evicted from L2, the updated value gets written to memory

# MESI Protocol States

| State | Valid | Modified | Copies may exist in other processors |
|---|---|---|---|
| Invalid | No | N.A. | N.A |
| Shared | Yes | No | Yes |
| Exclusive | Yes | No | No |
| Modified | Yes | Yes | No |

- **A modified line must be exclusive**
  - Otherwise, another processor which has the line will be using stale data
  - Therefore, before modifying a line, a processor must request ownership of the line

# MESI Protocol Example

- A four-processor shared-memory system implements MESI protocol
- For the following sequence of memory references, show the state of the line containing the variable X in each processor's cache after each reference is resolved
- Each processors start out with the line containing X invalid in their cache

| | **P0's state** | **P1's state** | **P2's state** | **P3's state** | **CVBs** |
|---|---|---|---|---|---|
| Initial State | *I* | *I* | *I* | *I* | 0000 |
| P0 reads X | *E* | *I* | *I* | *I* | 1000 |
| P1 reads X | *S* | *S* | *I* | *I* | 1100 |
| P2 reads X | *S* | *S* | *S* | *I* | 1110 |
| P3 writes X | *I* | *I* | *I* | *M* | 0001 |
| P0 reads X | *S* | *I* | *I* | *S* | 1001 |